АППАРАТНАЯ РЕАЛИЗАЦИЯ МОДУЛЕЙ ПАРАЛЛЕЛЬНОГО И ПОСЛЕДОВАТЕЛЬНОГО ПОДСЧЕТА КОНТРОЛЬНОЙ СУММЫ ДЛЯ ТРЕХ 32-БИТНЫХ СЛОВ

Рубцов И.Н., Новожилов И.В.
Научный руководитель: А.Н. Мальчуков
Томский политехнический университет
E-mail: inr2@tpu.ru

Введение

Для проверки целостности данных существуют различные методы подсчета контрольной суммы CRC. Классический (побитовый) алгоритм реализуется с помощью последовательного итерационного сдвига данных в регистре с обратной связью на один бит. Недостатком этого метода является низкая скорость работы. Особенностью табличного алгоритма является то, что при расчете контрольной суммы используется таблица с предвычисленными значениями на основе образующего полинома. При использовании данного метода используется большой объем памяти, т.к. требуется хранить предвычисленные данные. Матричный алгоритм работает так же, как и табличный, за исключением того, что вместо таблицы используется операция умножения выдвинутого вектора на матрицу по модулю 2. Подробное описание и анализ методов представлены в работе [3]. Основываясь на анализе алгоритмов вычисления CRC, выбран матричный алгоритм.

Далее приведено описание матричного алгоритма:

Шаг 1. Сложение по модулю 2 входного слова и сдвигового регистра. На первой итерации каждый бит сдвигового регистра содержит «1»;

Шаг 2. Результат сложения умножается на образующую матрицу;

Шаг 3. Полученный вектор записывается в сдвиговый регистр и является контрольной суммой;

Шаг 4. Для накопления контрольной суммы выполняются пункты 1-3 (пока не закончатся входные данные).

Задача

Задачей является реализация модуля на языке Verilog/VHDL, вычисляющего контрольную сумму для трех 32-битных слов на частоте 250 МГц.

Исходя из этого, существуют два пути возможного решения данной задачи: накопление контрольной сумму от каждого слова, либо обработка одного 96-битного слова.

Реализация модуля подсчета контрольной суммы CRC32 (с накоплением)

Данный модуль (Рис. 1) обрабатывает N 32-битных слов, последовательно накапливая контрольную сумму.Модуль Input reg выполняет роль входного регистра, который записывает входные данные по сигналу Valid_data = «1» и устанавливает сигналы busy = «1» и valid = «1». При Valid_data = «0» сигнал valid и busy сбрасываются в «0». Сигнал valid показывает истинность записанных данных, busy – свидетельствует о начале обработки данных. Модуль Crc32 – асинхронная комбинационная схема. В данном модуле происходит накопление контрольной суммы. Для реализации этого модуля применяется полином 0x04C11DB7.Output reg являются выходным регистром для данных и сигнала valid_crc (валидность контрольной суммы).

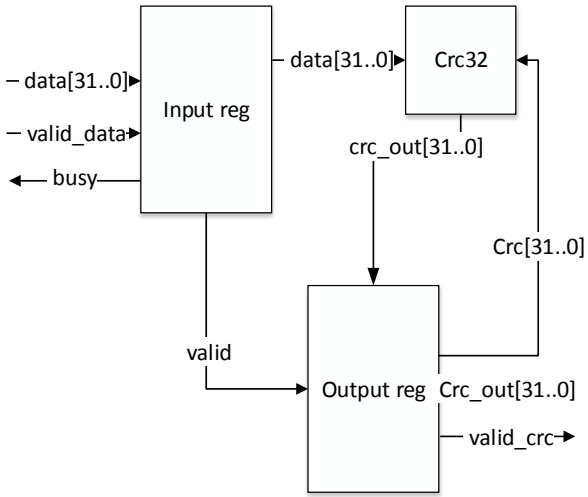

Рис. 1. Структурная схема модуля подсчета CRC32 матричным методом с накоплением

Если количество обработанных слов меньше чем заданный параметр, контрольная сумма передается обратно в блок Crc32.После накопления суммы итоговая контрольная сумма инвертируется и отправляется дальше(Crc_out). Valid_crc указывает на актуальность записанной в регистре контрольной суммы.

Реализация модуля подсчета контрольной суммы CRC32 (без накопления)

В данной реализации было принято решение по объединению трех 32-разрядных слов в одно 96-разрядное (Рис. 3). Таким образом можно заменить операцию сложения по модулю 2 регистра и входных данных на инверсию старших 32-ух битов входного слова. Такой подход позволит существенно увеличить производительность, но и увеличит образующую

матрицу в 3 раза, что повлияет на количество используемых ресурсов.

Методика тестирования и результаты работы

Для тестирования модуля использовалась программа на языке C++. Данная программа генерирует 2 файла. В первом содержится указанное число тестовых слов (слова сгенерированы случайным образом), а во втором соответствующие им контрольные суммы. Далее был написан тест, который читает файлы с данными и контрольными суммами.

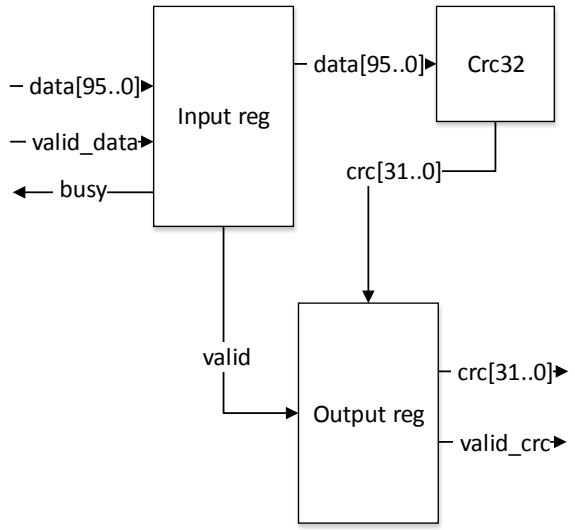

Рис. 2. Структурная схема модуля подсчета CRC32 матричным методом без накопления

Тестовые слова подаются на вход модуля, а полученные контрольные суммы сравниваются с тестовыми. Результат сравнения выводится в консоль.Как видно на Рис. 3., обработка одного тестового слова происходит за 1 такт. Тестирование подтвердило работоспособность модулей на требуемой частоте (250 МГц) и на завышенной частоте (400 МГц).

Заключение

В результате работы были реализованы два модуля для подсчета контрольной суммы CRC32. 96-битный модуль обрабатывает тройное слово за один такт, а 32-битный за 3 такта. 96-битный модуль использует большее количество вычислительных ресурсов, за счет чего достигается более высокая производительность.

- 32-битный модуль - 138 LUT, 102 FF;
- 96-битный модуль - 206 LUT, 131 FF.

(FF – D-триггер, LUT – таблица поиска).

Однако, 32-битный модуль является более гибким, т.к. имеет возможность подсчитывать контрольную сумму от любого заданного количества 32-битных слов. В то время как 96-битный модуль только от 96-битного слова.

Список литературы

1. Еремин В. В., Мальчуков А. Н. О применении блочно-ориентированного подхода к разработке устройств на ПЛИС [Электронный ресурс] // Вестник науки Сибири. Серия: Информационные технологии и системы управления. – 2011 – №. 1 – С. 379-381. – Режим доступа:http://sjs.tpu.ru/journal/issue/view/2/showToc/sect/4, свободный (дата обращения: 7.07.2016).

2. Мыцко Е. А., Мальчуков А. Н. Исследование программных реализаций алгоритмов вычисления CRC совместных с PKZIP, WINRAR, ETHERNET // Известия Томского политехнического университета. – 2013 – Т. 322 – №. 5. – С. 170-175

3. Мыцко Е. А., Мальчуков А. Н. Особенности программной реализации вычисления контрольной суммы CRC32 на примере PKZIP, WINZIP, ETHERNET [Электронный ресурс] // Вестник науки Сибири. Серия: Информационные технологии и системы управления. – 2011 – №. 1 – С. 279-282. – Режим доступа: http://sjs.tpu.ru/journal/issue/view/2/showToc/sect/4, свободный (дата обращения: 7.07.2016).

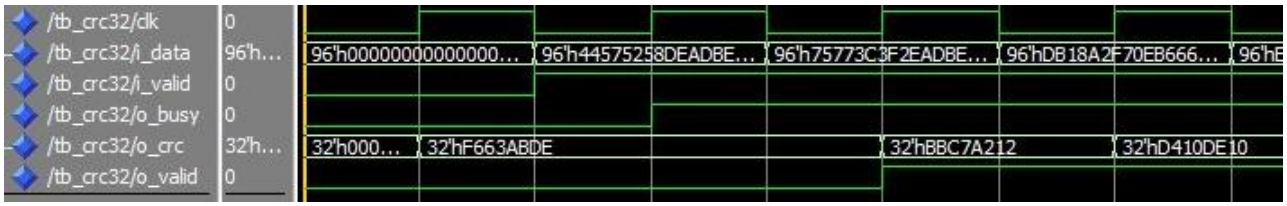

Рис. 3. Тестирование 96-битного модуля

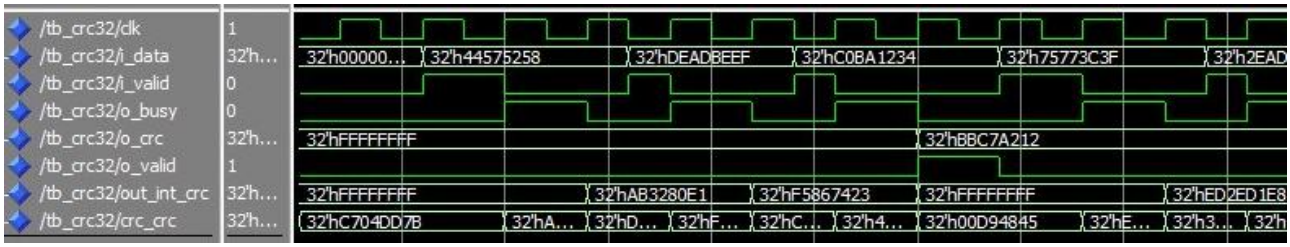

Рис. 4. Тестирование 32-битного модуля